Cultivating C Language Programming Competence and Innovating Teaching Models Under the Drive of Artificial Intelligence

Juan Bai*

Shaanxi University of Technology, Hanzhong, 723001, China
**Corresponding author: ajuan6535@163.com*

Abstract: *With the advancement of information technology, artificial intelligence (AI) is gradually playing a significant role in programming education. Traditional C language programming instruction suffers from insufficient personalized support and a lack of responsiveness to students' diverse needs, which limits the improvement of their programming competence. This paper explores how AI drives the transformation of C language programming education, with a focus on the development of personalized learning paths, intelligent feedback mechanisms, and adaptive assessment systems, as well as AI's support in cultivating programming thinking. The integration of AI enables dynamic evaluation and real-time feedback in C language instruction, which not only enhances students' programming skills but also fosters innovation and deeper thinking. The innovative teaching model proposed in this study provides new perspectives for programming education, facilitating a shift from traditional models to intelligent and personalized approaches.*

Keywords: *artificial intelligence; C language; programming competence; teaching model; personalized learning; adaptive assessment*

Introduction

Programming education is a core component of modern information technology education and has become an essential part of both basic and higher education. However, traditional C language instruction faces challenges such as fixed content, monotonous learning methods, and insufficient personalized support, making it difficult to meet students' diverse learning needs. With the development of artificial intelligence, AI has demonstrated great potential in programming education by offering customized learning experiences and instant feedback based on students' progress and individual differences. AI-driven innovation in C language teaching not only enables personalized content delivery but also enhances students' programming thinking and problem-solving abilities through intelligent evaluation and dynamic adjustment. This study aims to explore the application of AI in C language programming education, analyze its impact on the optimization of the teaching system and the enhancement of programming competence, and provide both theoretical and practical guidance for the future of programming education.

1. Integration of Artificial Intelligence and C Language Programming Education

1.1 The Role and Development of Artificial Intelligence in Programming Education

The widespread application of artificial intelligence (AI) in the field of education—particularly its deep integration into programming instruction—has become a key driving force for transforming teaching paradigms and enhancing educational quality. Traditional programming education models often suffer from fixed instructional content, delayed feedback, and limited consideration for individual differences, significantly hindering the systematic development of students' programming competence. In contrast, AI technology, with its data-driven, intelligent response and adaptive learning features, enables precise analysis of students' learning behaviors and dynamically generates personalized instructional content and learning paths based on their knowledge mastery, learning pace, and cognitive styles. In recent years, the application of AI in programming education has extended from intelligent teaching aids to more complex learning systems, such as adaptive learning platforms, code generation systems based on natural language processing, and automated error diagnosis and feedback mechanisms. These developments have significantly improved both teaching efficiency and the effectiveness of students' programming practice. Moreover, the incorporation of AI has shortened the time required for students to grasp programming languages and algorithmic thinking, effectively stimulating their learning initiative and creativity, and advancing programming education from "standardized instruction" to "intelligent guidance." With the continuous evolution of AI technologies, programming education is expected to undergo further reconstruction and innovation in areas such as instructional organization, resource allocation, and learning assessment [1].

1.2 Traditional Models and Challenges in C Language Programming Instruction

Conventional C language programming instruction primarily relies on classroom lectures and teacher-led guidance, with content typically delivered in a sequential manner based on standard textbooks. While this model ensures systematic acquisition of basic programming knowledge, it suffers from a lack of personalized learning support and fails to address differences among students in programming ability, learning pace, and depth of understanding. Many students, when learning C language, tend to resort to rote memorization of basic syntax and concepts, lacking deeper comprehension of programming logic and practical application skills. This issue becomes more pronounced when students are confronted with complex tasks such as program debugging or algorithm optimization, as traditional instruction often falls short in helping them tackle the diversity and challenges inherent in problem-solving. Furthermore, conventional assessment methods, which mainly rely on final exams and assignments, are static and do not provide timely feedback on students' learning progress, nor can they accurately identify weaknesses in programming thinking and hands-on practice. Therefore, implementing real-time evaluation and dynamic feedback has become a major challenge in current C language programming instruction.

1.3 Transforming C Language Teaching Models Through AI Technologies

With the rapid development of AI technology, the introduction of AI has brought about profound changes to the teaching model of C language programming. AI enables personalized learning experiences through precise data analysis and tailored learning path recommendations. For example, based on students' learning records and error patterns, AI can automatically recommend appropriate exercises and programming challenges, helping students enhance their skills at suitable levels of difficulty. AI tutoring systems can also detect code errors in real time and provide targeted feedback, thereby accelerating students' mastery of programming languages and deepening their understanding of mistakes. Moreover, AI can simulate programming environments to facilitate virtual hands-on practice, enhancing practical learning outcomes. For students with weak programming foundations, AI effectively provides supplemental learning content, helping them quickly bridge knowledge gaps. Additionally, AI can analyze large datasets of student performance to uncover potential problems in the teaching process and support educators in refining their instructional strategies and materials more accurately. Through these transformations, AI not only enhances the personalization and interactivity of C language instruction but also promotes the evolution of programming education toward more diversified and intelligent approaches [2].

2. AI-Based Mechanisms for Cultivating C Language Programming Competence

2.1 AI-Driven Personalized Learning Path Design

Artificial intelligence provides strong personalized learning support for C language programming education. By leveraging data analysis and machine learning capabilities, AI can track students' learning processes in real time, accurately identify weaknesses and growth potential in programming learning, and subsequently design customized learning paths for each student. AI systems analyze multidimensional data such as programming habits, error types, and problem-solving approaches to recommend the most appropriate learning content and exercises. In this process, students' progress and learning styles serve as the core basis for optimizing learning paths. For instance, students who progress rapidly may receive more challenging programming tasks to further enhance their creativity and programming skills, while those encountering difficulties are provided with more fundamental content along with real-time guidance and prompts. This personalized learning path design not only improves learning efficiency but also enhances students' interest in programming, avoiding the "one-size-fits-all" approach of traditional instruction and enabling each student to improve at their own optimal pace.

2.2 Construction of Intelligent Feedback and Adaptive Assessment Systems

In programming education, feedback and assessment are critical to improving learning outcomes. AI-powered intelligent feedback systems and adaptive assessment mechanisms offer real-time and efficient solutions for C language instruction. By analyzing students' submitted code, AI algorithms can instantly detect programming errors, logical flaws, and areas for optimization, providing detailed diagnostics and corrective suggestions. Unlike traditional assessment systems, AI-driven evaluation not only judges code accuracy but also offers personalized advice based on students' programming habits and cognitive patterns. For example, if a student repeatedly makes a specific type of error, the system prioritizes a review of the relevant concepts and recommends targeted exercises. This adaptive assessment system dynamically adjusts its evaluation criteria and feedback content, ensuring timely and precise guidance at every learning stage. Furthermore, AI systems can analyze students' long-term learning trajectories to generate comprehensive competence assessments and recommend focus areas for future study, thereby supporting systematic and phased improvement in C language programming ability.

2.3 AI-Supported Strategies for Developing Programming Thinking

Cultivating programming thinking is a central objective of C language education. AI technology assists students in developing logical reasoning and problem-solving skills through multiple approaches, facilitating deeper and broader growth in programming thinking. Intelligent programming environments powered by AI can simulate various coding scenarios, helping students understand and solve real-world problems and thereby enhancing the flexibility and creativity of their programming mindset. For example, AI-based virtual coding assistants provide real-time suggestions during code writing, guiding students in efficient program design, algorithm optimization, and code refactoring to reinforce proper coding practices. Throughout this process, students gain not only a grasp of programming syntax and rules but also the ability to approach problems from multiple perspectives, improving their capacity to solve complex programming tasks. Additionally, AI encourages exploratory learning by offering programming challenges that stimulate creative thinking and promote the transition from concrete problem-solving to abstract reasoning. In this way, AI not only advances students' coding skills but also accelerates the development of programming thinking, enabling the flexible application of knowledge and techniques in real-world programming contexts [3].

3. AI-Driven Innovation in C Language Teaching Models

3.1 Interactive C Language Classroom Design Supported by AI Technology

3.1.1 Real-Time Dynamic Feedback and Personalized Interaction

AI technology enables real-time monitoring of students' learning progress and comprehension in class, providing immediate feedback. Teachers can access detailed data on students' programming tasks through AI systems, such as the types of coding errors, debugging time, and problem-solving approaches. Based on this data, AI systems can automatically generate questions or programming tasks to help students better understand and master C language concepts. This intelligent feedback allows instructors to adjust teaching strategies in real time, ensuring that each student receives personalized learning support. AI can also recommend individualized exercises or learning resources based on students' performance, effectively promoting autonomous learning beyond the classroom. This form of personalized interaction not only improves learning outcomes but also fosters students' interest in programming and enhances their motivation to learn.

3.1.2 Support for Diversified Learning Methods

AI technology expands the C language classroom beyond traditional lectures and after-class assignments. Through virtual experiments, code simulations, and simulated programming environments, AI provides students with various forms of interactive learning experiences. For example, students can use virtual labs to practice coding without time and location constraints, or simulate complex programming tasks to explore the intricacies of real-world applications. Such diversified learning support not only enriches instructional content but also enables students to discover learning styles that best suit them, thereby improving learning efficiency. AI can also track students' learning behaviors, analyze their preferences in real time, and offer feedback to teachers, enabling more flexible and personalized instructional strategies [4].

3.1.3 Intelligent Classroom Management and Personalized Teaching

The integration of AI technology enhances classroom management by making it more efficient and targeted. Teachers can access detailed reports on each student's learning progress, class participation, and assignment completion through AI systems. These systems help instructors monitor students' progress and difficulties in learning C language and provide clear feedback for instructional adjustment. This intelligent classroom management improves teaching efficiency and ensures smoother learning processes. AI analysis also allows for the timely identification of students' problems and the provision of personalized guidance and support. Overall, AI-enhanced classroom design makes C language instruction more precise and effective, significantly boosting interactivity and the overall student learning experience.

3.2 Learning Behavior Analysis and Teaching Optimization Based on Big Data

3.2.1 Learning Behavior Data Collection and Analysis

AI technology can collect behavioral data from students' programming activities in real time, including coding speed, error frequency, and problem-solving methods. These data help create personalized learning profiles for each student and allow teachers to identify individual strengths and weaknesses in C language learning. By analyzing this behavioral data, AI can detect frequent issues with certain concepts and help instructors refine their content or methods accordingly. For example, if a group of students repeatedly struggles with pointers or memory management, the system alerts the teacher to reinforce those areas. Through big data analysis, instructors can monitor learning progress accurately and detect teaching inefficiencies, enabling them to adjust strategies and improve overall instructional quality [5].

3.2.2 Dynamic Teaching Adjustment and Personalized Resource Recommendation

AI systems, powered by big data analysis, can automatically adjust instructional content and delivery methods to meet diverse student needs. Based on students' learning behaviors and progress, the system dynamically recommends learning materials suited to their current level. For students who advance quickly, AI offers more challenging programming tasks to deepen their understanding of C language; for those with slower progress, it provides foundational exercises and supplementary materials to help reinforce their basics. By dynamically adjusting content and resources, AI enables more flexible and personalized teaching, avoiding the limitations of a "one-size-fits-all" model.

3.2.3 Teaching Effectiveness Evaluation and Continuous Optimization

Another key application of big data is the evaluation and ongoing refinement of teaching effectiveness. AI can analyze students' grades, classroom performance, and completion of programming assignments to generate comprehensive reports on instructional outcomes. These analytics reveal which teaching methods are most effective, which topics need further emphasis, and which resources are most helpful to students. Based on these insights, teachers can adjust their strategies, improve lesson design, and refine assessment standards to achieve continuous teaching optimization. The big data–driven evaluation mechanism not only enhances instructional efficiency but also ensures ongoing improvement in student learning outcomes.

3.3 AI-Enhanced Assessment and Advancement Mechanisms for Programming Competence

3.3.1 Construction of a Multidimensional Competency Assessment System

The AI-enabled programming assessment system evaluates students' skills across multiple dimensions. Unlike traditional methods that focus mainly on code correctness, AI systems assess code quality, execution efficiency, algorithm design, and problem-solving strategies, offering comprehensive performance feedback. By dynamically analyzing students' programming assignments, AI can evaluate innovation in design and depth of problem-solving, helping students identify both strengths and areas for improvement. This multidimensional assessment more accurately reflects students' programming thinking and overall competence, providing targeted recommendations for further learning.

3.3.2 Adaptive Advanced Learning Pathways

AI technology allows for non-linear, adaptive progression in programming competence development. By continuously monitoring students' performance in programming activities, AI systems can tailor advanced learning paths. For students who demonstrate strong capabilities after mastering the basics, AI recommends more challenging tasks; for those encountering persistent difficulties, the system offers additional reviews and support materials to ensure steady progress. AI-driven progression ensures that students engage with the most suitable programming tasks at the most appropriate stages, thereby maximizing learning effectiveness [6].

3.3.3 Continuous Monitoring and Feedback Mechanism

AI's ongoing monitoring and feedback mechanism tracks student performance throughout the learning process to ensure steady improvement in programming skills. By observing each phase of coding activities, AI systems can identify learning progress in real time and deliver timely guidance. This dynamic feedback helps students clarify their goals and provides prompt support when difficulties arise, preventing stagnation. The AI-powered continuous feedback mechanism enables comprehensive, accurate evaluation of programming competence and motivates students to maintain consistent learning and progress.

Conclusion

This study explored the application of artificial intelligence in C language programming education and its impact on the innovation of teaching models. AI technology not only enables the design of personalized learning paths but also facilitates the comprehensive improvement of students' programming competence through intelligent feedback and adaptive assessment systems. In addition, AI supports the cultivation of programming thinking, guiding students from basic coding skills training toward higher-order logical reasoning and problem-solving abilities. Looking ahead, as AI technology continues to evolve, C language programming education will encounter further opportunities for innovation. Future research may delve deeper into the application of AI in more complex programming environments, integrating machine learning and AI technologies to develop more precise programming assessment systems. It may also explore how to combine technologies such as virtual reality and augmented reality with programming instruction to enhance students' practical skills. AI-empowered programming education not only addresses limitations in traditional instruction but also promotes the personalization and intelligent transformation of education, driving broader educational reform.

Funded Projects

Shaanxi Province "14th Five-Year Plan" Education Science Research Project, 2024 (SGH24Y2189);

Shaanxi University of Technology "Curriculum-Based Ideological and Political Education" Teaching Reform Project (KCSZ2419);

Curriculum-Based Ideological and Political Education Demonstration Course Construction Project for Programming Fundamentals (421/131050379);

Shaanxi University of Technology 2024 University-Level Smart Course Development Project.

References

[1] Wang Jinling, Huang Qingqing, and Chai Wandong. "Application of AIGC-Based Teaching Case Models in C Language Instruction." Journal of Chifeng University (Natural Science Edition) 41.05(2025): 107–110.
[2] Xiao Yao. "A Study on the Application of Artificial Intelligence Systems in C Language Programming Teaching." Science & Technology Information 23.08(2025): 214–217.
[3] Chen Jianzeng. "Research on the Design Strategy of C Language Programming Experiment Teaching Systems Based on AI Technology." Information & Computer (Theoretical Edition) 36.07(2024): 46–48.
[4] Chen Hong. "Exploration of Teaching Reform in the Public Course 'C Language Programming' in the Era of Artificial Intelligence." Science & Technology Information 22.04(2024): 194–197.
[5] Liao Bin, Cai Qilei, and Huang Rongrong. "Teaching Reform of the C Language Programming Course in the Context of Informatization and Intelligence." Academy 15.18(2022): 33–35.
[6] Zhang Hongmei, Lu Bangding, and Wang Fang. "School-Based Practice of the Integrated C

Language Curriculum under the Background of Artificial Intelligence." Vocational Education (Mid-Month Edition) 20.11(2021): 11–14.